Listing of the Claims:

1.   (original)   A method for joining an inner table and an outer table in a database in response to a query statement having an inclusion operator, said method comprising:

(a)   providing an inner table;

(b)   providing an outer table having zero or more records for inclusion with said inner table;

(c)   transposing said outer table and said inner table to form a right table and a left table, respectively;

(d)   creating a hash table from said left table;

(e)   obtaining a hash value from said right table;

(f)   probing said hash table with a said hash value from said right table to determine if said hash value matches a value in said hash table; and

(g)   if said outer hash value matches said value in said hash table, then evaluating the actual values and, if the join condition is satisfied, including said row of said outer table in a result.

2.   (original)   The method of claim 1, wherein scanning said inner table is accomplished one row at a time per processor.

3.   (original) The method of claim 1, wherein said step of obtaining a hash value from said right table includes calculating a hash value.

4.   (original) The method of claim 1, wherein said step of obtaining a hash value from said right table includes retrieving a pre-calculated hash value from a row header.

5. (original) A method for joining an inner table and an outer table in a database in response to a query statement having an exclusion operator, said method comprising:

(a) providing an inner table;

(b) providing an outer table having zero or more records for inclusion with said inner table;

(c) transposing said outer table and said inner table to form a right table and a left table, respectively;

(d) creating a hash table from said left table;

(e) obtaining a hash value from said right table;

(f) probing said hash table with a said hash value from said right table to determine if said hash value matches a value in said hash table;

(g) if said right hash value do not match any value in said hash table, then including said row from said right table in a result.

6. (original) The method of claim 5, wherein scanning said outer table is accomplished one row at a time per processor.

7. (original) The method of claim 5, wherein said step of obtaining a hash value from said right table includes calculating a hash value.

8. (original) The method of claim 5, wherein said step of obtaining a hash value from said right table includes retrieving a pre-calculated hash value from a row header.

9. (original) A database system having a database with at least two tables for responding to SQL statements having an inclusion join condition that designate an inner table and an outer table within said database, said database system comprising:

system memory, said system memory constructed and arranged to store a hash table;

a processor operative with said system memory, said processor constructed and arranged to generate a hash table from said inner table, said processor further constructed and arranged to scan at least one row of said outer table, obtain an outer hash value, and compare said outer hash value to a hash value in said hash table to determine if said values match and, if so, evaluating actual values from said rows to determine if a join condition is satisfied and, if so, then adding said row from said outer table to a result.

10. (original) A database system having a database with at least two tables for responding to SQL statements having an exclusion join condition that designate an inner table and an outer table within said database, said database system comprising:

system memory, said system memory constructed and arranged to store a hash table;

a processor operative with said system memory, said processor constructed and arranged to generate a hash table from said inner table, said processor further constructed and arranged to scan at least one row of said outer table, obtain an outer hash value, and compare said outer hash value to a hash value in said hash table to determine if said values match and, if not, then adding said row from said outer table to a result.

11. (original) A database system for accessing a database in response to a SQL statement having an inclusion condition, said database system comprising:

a massively parallel processing system comprising:

one or more nodes;

a plurality of CPUs, each of the one or more nodes providing access to one or more CPUs;

a plurality of virtual processes each of the one or more CPUs providing access to one or more virtual processes;

each virtual process configured to manage data stored in one of a plurality of data-storage facilities;

a parsing engine comprising:

a session control, which controls access to the plurality of virtual processes;

a parser, which interprets and optimizes a command for access to a database to produce a series of steps to be performed; and

a dispatcher, which controls the sequence of the series of steps and determines which of the plurality of virtual processes will perform each of the series of steps;

where the parser includes:

an optimizer which optimizes a plan for executing the series of steps;

where the optimizer includes:

an inclusion hash join capability; and

capability to recognize an inclusion merge join and replace with said inclusion hash join having an inner table and an outer table, wherein said inner table and said outer table are assigned as a left table and a right table, respectively, and

said left table is hashed into a hash table and hash values from said right table are used to probe said hash table to determine if an inclusion condition is satisfied for a row of said right table and, if so, then evaluating a join condition and, if said evaluated indicates said inclusion condition is satisfied, then including said row of said right table in a result.

12.   (original)   A database system for accessing a database in response to a SQL statement

having an exclusion condition with an inner and an outer table, said database system comprising:

a massively parallel processing system comprising:

one or more nodes;

a plurality of CPUs, each of the one or more nodes providing access to one or more

CPUs;

a plurality of virtual processes each of the one or more CPUs providing access to

one or more virtual processes;

each virtual process configured to manage data stored in one of a plurality of data-

storage facilities;

a parsing engine comprising:

a session control, which controls access to the plurality of virtual processes;

a parser, which interprets and optimizes a command for access to a database to

produce a series of steps to be performed; and

a dispatcher, which controls the sequence of the series of steps and determines which

of the plurality of virtual processes will perform each of the series of steps;

where the parser includes:

an optimizer which optimizes a plan for executing the series of steps;

where the optimizer includes:

an exclusion hash join capability; and

capability to recognize an exclusion merge join and replace with said exclusion hash

join wherein said inner and outer tables are assigned to a left table and a right table,

respectively, said left table is hashed into a hash table and a row of said right table is hashed

to obtain a hash value that used to probe said hash table to determine if said hash values match and, if no hash values match, then including said row of said right table in a result.